|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ ***ИУК «Информатика и управление»\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_ ***ИУК5 «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

***Калькулятор с расширенными возможностями***

по дисциплине ***Системное программирование***

Студент гр.\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2021

Калужский филиал   
федерального государственного бюджетного образовательного учреждения высшего образования

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»   
(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУК5\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(Е.В. Вершинин)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине ***Системное программирование***

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, индекс группы)

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы)

График выполнения проекта: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

***1. Тема курсового проекта***

***Калькулятор с расширенными возможностями***

***2. Техническое задание***

*Разработать калькулятор с использованием функций Windows API для вычисления арифметических,тригонометрических выражений ,а также нахождение корней линейных и квадратных уравнений* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***3. Оформление курсового проекта***

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О.)

Задание получил\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г.

(подпись) (Ф.И.О.)

Примечание:

Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**СОДЕРЖАНИЕ**

1. Техническое задание…………………………………………………………5-9
   1. Наименование…………………………………………………….…….5
   2. Основание для разработки……………………………………….……5
   3. Исполнитель……………………………………………………………5
   4. Цель разработки…………………………………………………..……5
   5. Содержание работы……………………………………………………6
      1. Задачи, подлежащие решению………………………………….…6
      2. Требования к архитектуре АСОИ…………………………………7
      3. Требования к составу программных компонентов……………….7
      4. Требования к прикладным программам…………………………..7
      5. Требования к входным/выходным данным…………………..…..7
      6. Требования к временных характеристикам…………………..…..8
      7. Требования к составу технических средств………………..……..8
   6. Этапы разработки……………………..………………………………..8
   7. Техническая документация, предъявляемая по окончании работы..8-9
   8. Дополнительные условия……………………..………………………..9
2. Научно-исследовательская часть……………………………………...……10-19
   1. Постановка задачи проектирования…………………………..……..10
   2. Описание предметной области………………………………..…….10-11
   3. Анализ аналогов и прототипов……………………..……………12-16
   4. Перечень задач, подлежащих решению в процессе разработки….16
   5. Обоснование выбора инструментов и платформы для разработки……………………………………………………..……...16-19
3. Проектно-конструкторская часть…………………………………………..20-33
   1. Разработка структуры приложения…………………………………20-22
   2. Используемые функции Win32API………………………………….22-27
   3. Разработка алгоритмов приложения…………………………….…..28-31
   4. Разработка интерфейса взаимодействия пользователя с системой………………………………………………………………32-33
4. Проектно-технологическая часть…………………………………………..34-49
   1. Тестирование и отладка макета рабочей программы………….….34-45
   2. Разработка руководства пользователя и руководства администратора……………………………………………………...45-49

Заключение………………………………………………………….…………50

Литература……………………………………………………………………..51

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

**1.1 Наименование**

Калькулятор с расширенными возможностями на основе Windows API.

**1.2 Основание для разработки**

Калькулятор с расширенными возможностями незаменимый помощник для студентов и инженеров, позволяющий производить вычисления начиная с элементарной линейной математики и заканчивая расчетами тригонометрических функций, логарифмов, факториалов, решений квадратных уравнений в поле комплексных чисел, матриц.

Но не все функции есть в инженерных калькуляторах. Данное же приложение сможет рассчитывать линейные уравнения, квадратные уравнения.

**1.3 Исполнитель**

Студент группы ИУК5-42Б Гусько Г.А.

**1.4 Цель разработки**

Целью курсовой работы является формирование практических навыков по разработке и реализации программного приложения с использованием интерфейса прикладного программирования (АРI) операционных систем.

Задачи проектирования:

1. овладение первичными навыками ведения научно-исследовательской, проектной и производственно-технологической деятельности, развитие творческих способностей индивидуально для каждого студента;
2. подготовка к выполнению выпускной квалификационной работы;
3. усвоение методов грамотного ведения, оформления и редактирования технической документации.

Целью разработки является изучение принципов системного программирования и работы WinAPI функций, и разработка калькулятора с богатым арсеналом возможностей для математических расчетов.

**1.5. Содержание работы**

**1.5.1. Задачи, подлежащие решению:**

исследование существующих калькуляторов с расширенными возможностями;

изучение принципов системного программирования и работы WinAPI функций;

реализация востребованных вычислительных функций калькулятора для простых и сложных расчётов;

проведение тестирования реализованных функций;

1. подготовка расчетно-пояснительной записки и графических листов;
2. подготовка презентации и речи для защиты курсовой работы;
3. защита курсовой работы.

**1.5.2. Требования к архитектуре АСОИ**

К архитектуре предъявляются следующие требования:

приложение должно представлять собой многопоточное приложение. Работа пользовательского интерфейса должна быть реализована в основном потоке, а все остальные функции калькулятора в отдельных потоках;

каждая функция(калькулятора), должна быть представлена в виде отдельного класса.

**1.5.3. Требования к составу программных компонентов**

Программный комплекс должен состоять из следующих программных компонентов:

1. исполняемый файл PE формата с расширением .exe, реализующий калькулятор.

**1.5.4. Требования к прикладным программам**

Для работы программного комплекса необходимы:

1. Microsoft Windows 8/10;
2. Microsoft Visual Studio 2019;
3. минимальный набор драйверов, обеспечивающих   
   работоспособность ПК.

**1.5.5. Требования к входным/выходным данным**

Входные данные:

1. Любые числа, представленные в виде вещественного типа данных, положительные и отрицательные.

Выходные данные:

1. Числа вещественного типа данных, положительные и отрицательные.

**1.5.6. Требования к временным характеристикам**

Требования к временным характеристикам программы   
не предъявляются.

**1.5.7. Требования к составу технических средств**

Для функционирования системы необходимы:

* процессор: любой 32-х или 64-х битный процессор;
* RAM: 1 Гбайт;
* HDD: 50 Мбайт;
* видеокарта: поддержка Microsoft DirectX 9;
* монитор;
* клавиатура;
* мышь.

**1.6. Этапы разработки**

исследование существующих калькуляторов с расширенными возможностями;

изучение принципов системного программирования и работы WinAPI функций;

выбор актуальных функций для проведения сложных расчетов;

реализация актуальных функций

проведение тестирования реализованных функций.

**1.7. Техническая документация, предъявляемая по окончании работы**

По окончанию работы предъявлена расчетно-пояснительная записка в состав которой входят:

* техническое задание;
* научно-исследовательская часть;
* проектно-конструкторская часть;
* проектно-технологическая часть.

Также должна быть предоставлена графическая часть работы, выполненная формате А1 на 2 листах, в которую входят:

* демонстрационные чертежи;
* алгоритмические схемы.

**8. Дополнительные условия**

язык программирования С/С++;

использование Windows API функций для реализации функционала калькулятора;

тип приложения – оконное.

# 2. Научно-исследовательская часть

## **2.1 Постановка задачи проектирования**

Задачей проектирования является формирование практических навыков по работе и реализации программного приложения с использованием интерфейса прикладного программирования (API) операционных систем.

Необходимо разработать программное обеспечение, позволяющее оперативно и удобно выполнять математические выражение, производить расчёт линейных и квадратных уравнений.

## **2.2. Описание предметной области**

**Калькулятор** – это [электронное](https://wiki2.org/ru/Электроника) [вычислительное устройство](https://wiki2.org/ru/Вычислительное_устройство) для выполнения [операций над числами](https://wiki2.org/ru/Операции_над_числами) или [алгебраическими формулами](https://wiki2.org/ru/Алгебраическая_функция).

Есть разные виды калькуляторов в зависимости от функционала:

**А) Физические:**

* простые калькуляторы;
* специализированные калькуляторы;
* простые научные калькуляторы;
* современные научные калькуляторы.

**Б) Программные**

**Физические калькуляторы:**

**Простые калькуляторы:**

Имеют минимальное число функций – сложение, вычитание, умножение, деление, возведение в квадрат, извлечение корня квадратного, процент, память на одно число. Недостаток: нет тригонометрических функций (sin, cos, tg, ctg, arcsin, arcos, arctg) и запись вычислительных выражений только линейная.

**Специализированные калькуляторы**:

К ним относятся финансовые калькуляторы, статистические, бухгалтерские, проекционные, печатающие и т.п. Недостаток: нет тригонометрических функций (sin, cos, tg, ctg, arcsin, arcos, arctg) и запись вычислительных выражений только линейная, а также есть ненужные специализированные кнопки.

**Простые научные калькуляторы**

Имеют большее количество математических функций, в том числе, тригонометрические. Однострочный линейный порядок ввода, без учёта скобок и т.д.

**Современные научные калькуляторы**

Они содержат большое количество функций, необходимых для вычислений, в том числе и тригонометрические функции (sin, cos, tg, ctg, arcsin, arcos, arctg). Отдельные модели позволяют проводить статистические расчёты, вычисление интегралов и дифференциалов, операции с векторами и матрицами и т.д. Порядок ввода выражения нелинейный, т.е. можно ввести выражение в таком же виде, как оно записано на листочке. Есть режим правки и история вычислений, что позволяет не вводя заново, вернуться для исправления.

**Программные калькуляторы**

ПО калькулятора представляет собой калькулятор, который был реализован в виде компьютерной программы, а не в качестве физического аппаратного устройства.

Существует много различных аналогов в зависимости от нужного функционала и ОС.

Калькуляторы есть для следующих ОС: Windows XP, Windows Vista, Windows 7, Windows 10, Mac OS X и т.д.

Как правило, ПО калькулятора характеризуется следующим:

* есть небольшой набор относительно простых операций;
* выполняет короткие процессы, не требующие интенсивных вычислений;
* не принимает большие объемы входных данных.

## **2.3 Анализ аналоговых прототипов.**

Для того чтобы помочь пользователям, столкнувшимся с определенными трудностями в математических вычислениях, было разработано огромное количество самых разнообразных программ:

**CrossGL Surface Calculator**, простой обычный калькулятор. Единственное, что предлагает CrossGL, это отличный хорошо известный интерфейс. CrossGL Surface Calculator идентичен любому калькулятору, который можно увидеть в лавке канцтоваров, что является его большим преимуществом. CrossGL Surface Calculator включает в себя преобразователь валют. Преобразователь валют включает множество опций конфигурации. По умолчанию, он преобразовывает любую валюту в американские доллары. Тем не менее, приложение можно настроить так, чтобы он преобразовывал практически любую валюту. Интерфейс представлен на рисунке 1.

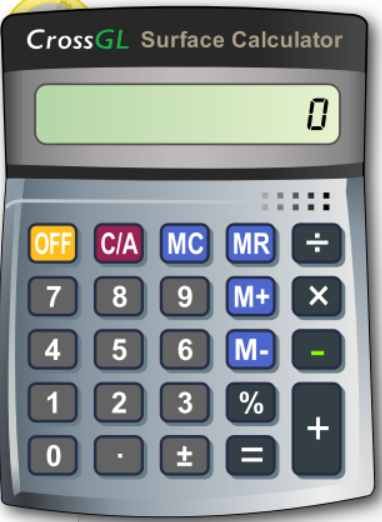


Рисунок 1 Интерфейс CrossGL Surface Calculator.

**Достоинства:**

* понятный интерфейс;
* преобразователь валют;
* необходимый функционал.

**Недостатки:**

* не включает дополнительные опции;
* запись вычислительных выражений только линейная;
* не может решать уравнения (линейные и квадратные).

**MyCalc2** - это удобный в использовании научный калькулятор, имеющий ряд функций, которые превращают его в одну из лучших разработок в своей категории, поскольку он идеально сочетает простоту использования с очень интересными функциями. Интерфейс представлен на рисунке 2.



Рисунок 2 Интерфейс MyCalc2.

**Достоинства:**

* различные методы расчета (статистический, временной, финансовый);
* персонализация внешнего вида;
* редактор для создания формул;
* память о последних выполненных операциях.

**Недостатки:**

* не включает дополнительные опции(тригонометрию);
* не может решать уравнения (линейные и квадратные).

**ESBCalc** включает все, что требуется от продвинутого калькулятора. Этот калькулятор имеет очень простой интерфейс, где отображаются все классические кнопки, включенные в вычислительном программном обеспечении, а также расширенные функции. Но ESBCalc также показывает возможность скопировать результат в буфере обмена, чтобы легко вставить его где-нибудь в другом месте.

И, конечно же, он способен решать тригонометрические, гиперболические и логарифмические вычисления на десятое и двоичное основание и натуральное.

Более того, ESBCalc позволяет просматривать историю расчетов и настроить количество десятичных, которых мы хотим использовать в наших вычислениях, чтобы настроить результат в максимально возможной степени. Интерфейс представлен на рисунке 3.

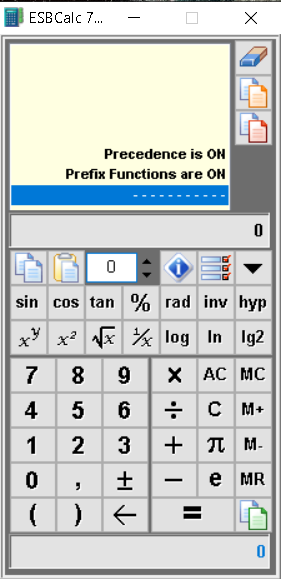


Рисунок 3 Интерфейс ESBCalc.

**Достоинства:**

* простой интерфейс;
* возможность скопировать результат в буфере обмена;
* способен решать тригонометрические, гиперболические и логарифмические вычисления;
* возможность просмотреть историю расчётов.

**Недостатки**:

* запись вычислительных выражений только линейная;
* не может решать уравнения (линейные и квадратные).

Калькулятор Windows, это простой, но мощный калькулятор, который включает такие режимы, как “Стандартный”, “Научный”, “Программист”, а также конвертер валют. Это идеальный инструмент для вычисления суммы счета, преобразования единиц измерений для рецепта или другого проекта, а также для решения сложных задач по математике, алгебре или геометрии. Журнал калькулятора позволит легко проверить, правильно ли были введены цифры. Интерфейс представлен на рисунке 4

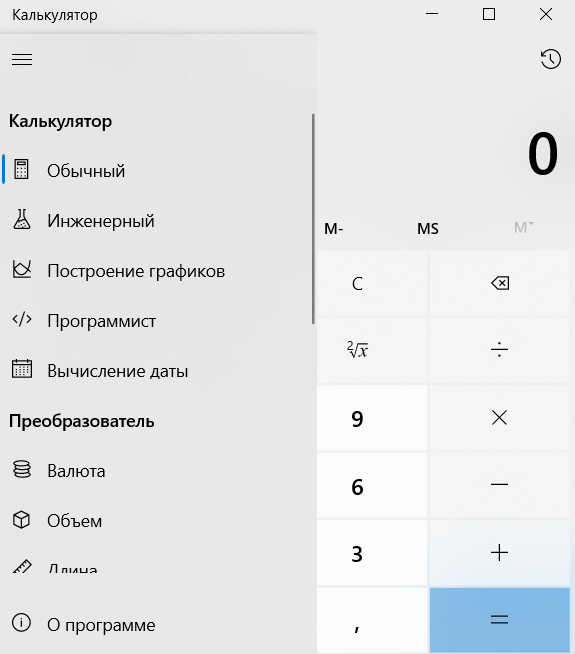


Рисунок 4 Интерфейс калькулятора Windows.

**Достоинства:**

* простой интерфейс;
* большое количество функций и возможностей;
* возможность просмотреть историю расчётов.

**Недостатки**:

* запись вычислительных выражений только линейная;
* не может решать уравнения (линейные и квадратные).

Таким образом каждый калькулятор имеет ряд достоинств и недостатков, одни имеют удобный интерфейс вкупе с широким набором возможностей для расчётов, другие имеют только необходимые функции. Поэтому целью данной работы является создание приложения, позволяющее проводить не только простые расчёты, тригонометрические функции, но и решать уравнения (линейные и квадратные) с удобным интерфейсом

## **2.4 Перечень задач, подлежащих решению в процессе разработки.**

1. разработка пользовательского интерфейса с использованием WinAPI;
2. разработка базового функционала калькулятора;
3. разработка функций для решения уравнений;
4. привязка функций калькулятора к пользовательскому интерфейсу;
5. тестирование разработанного приложения;
6. исправление выявленных ошибок.

## **2.5 Обоснование выбора инструментов и платформы для разработки.**

Платформой для разработки была выбрана Windows 10, которая на данный момент является самой распространенной ОС в мире в соответствии с исследованием от NetMarketShare. Статистика приведена на рисунке 5

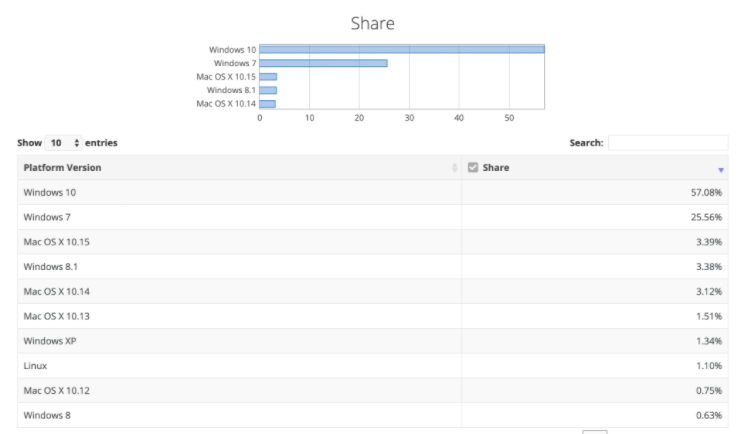


Рис 5 Статистика использования операционных систем в мире.

Windows - семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении.

Windows представляет гибкий инструмент в виде библиотек и функций, основные платформы приложений каждая из которых имеет ряд преимуществ:

* Универсальная платформа Windows(UWP)-предоставляется система общих типов, интерфейсы API и модель приложений для всех устройств под управление Windows 10;
* Win32/Win 64 – исходные платформы для нативных Windows приложений C/C++, которым требуется прямой доступ к Windows и оборудованию.

В качестве языка программирования был выбран язык С++. Язык успешно позволяет создать как простые приложения и утилиты, так и структурно сложные проекты. С++ является ISO-стандартизированным языком программирования

C++ - компилируемый статически типизированный язык программирования общего назначения. Поддерживая разные парадигмы программирования, сочетает свойства как высокоуровневых, так и низкоуровневых языков. Позволяет работать с библиотеками. Область его применения включает операционные системы, прикладные программы, драйверы устройств, приложения для встраиваемых систем, высокопроизводительные серверы, а также развлекательные приложения.

Средой проектирования, позволяющей реализовать подобное программное обеспечение, была выбрана Microsoft Visual Studio 2019 представлена на рисунке 6. Среда разработки Visual Studio представляет собой полный набор средств для разработки и создания веб приложений, настольных приложений и мобильных приложений.Visual C++ использует единую интегрированную среду разработки (IDE), которая позволяет совместно использовать средства и упрощает создание решений на базе нескольких языков.

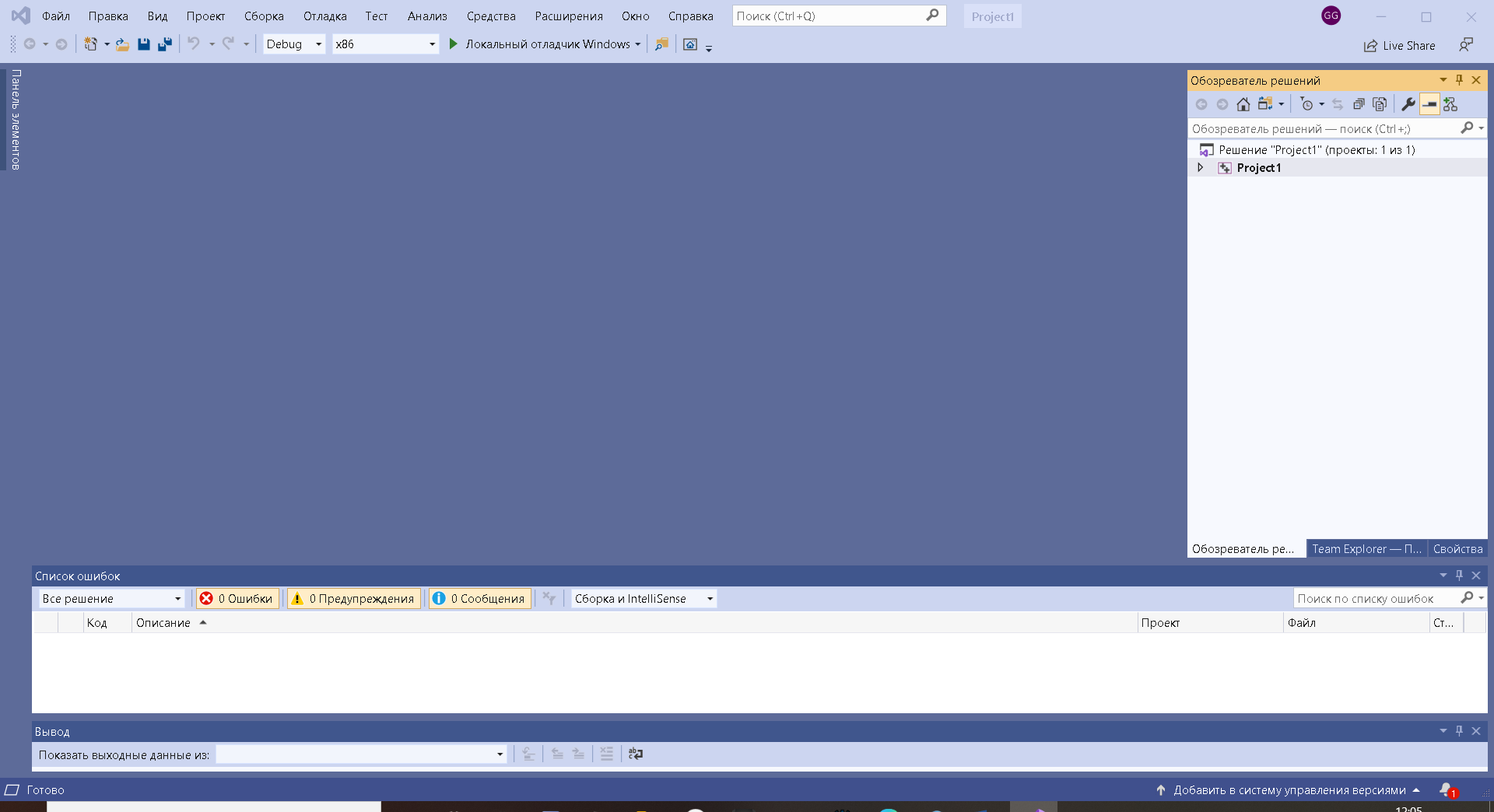


Рис 6 Интерфейс Microsoft Visual Studio 2019.

Преимущества выбранной среды программирования:

* удобство интерфейса;
* сравнительно не высокие требования к системе;
* удобное средства отладки;
* возможность хранения данных в облаке;
* поддержка множества языков программирования;
* наличие бесплатной редакции Community;
* встроенный контроль за выполнением многопоточного кода.

**3.ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ**

**3.1 Разработка структуры приложения**

Структурно разрабатываемое приложение можно разделить на следующие части (например: Рисунок 7, Рисунок 8 - UML диаграмма вариантов использования), графический пользовательский интерфейс, алгоритмы математических функций.

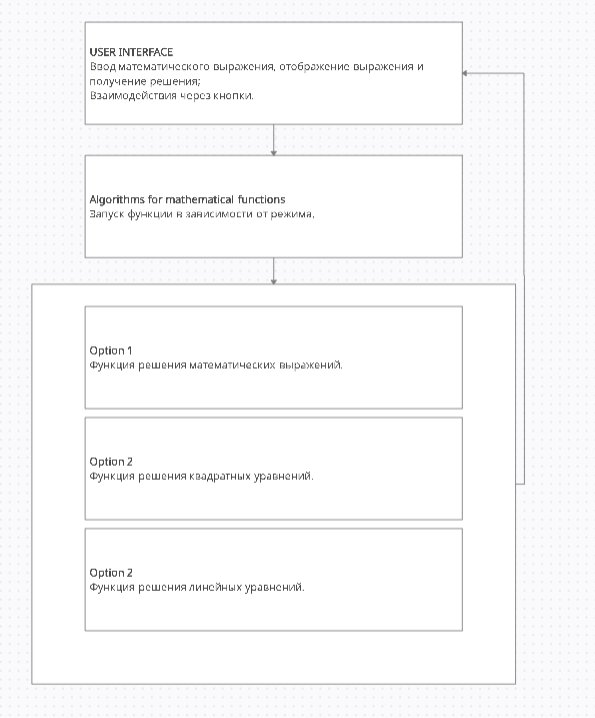
Графический пользовательский интерфейс необходим для удобного взаимодействия пользователя с функционалом разрабатываемого приложения «Калькулятор». Он должен давать возможность быстро и удобно через кнопки набирать математические выражения, а также отображать полученное решение в предусмотренном поле.

Алгоритмы математических функций требуются для проведения математических операций. Должны быть разбиты на три функции:

1)Для обычных расчётов;

2)Для расчётов квадратных уравнений;

3)Для расчётов линейных уравнений.



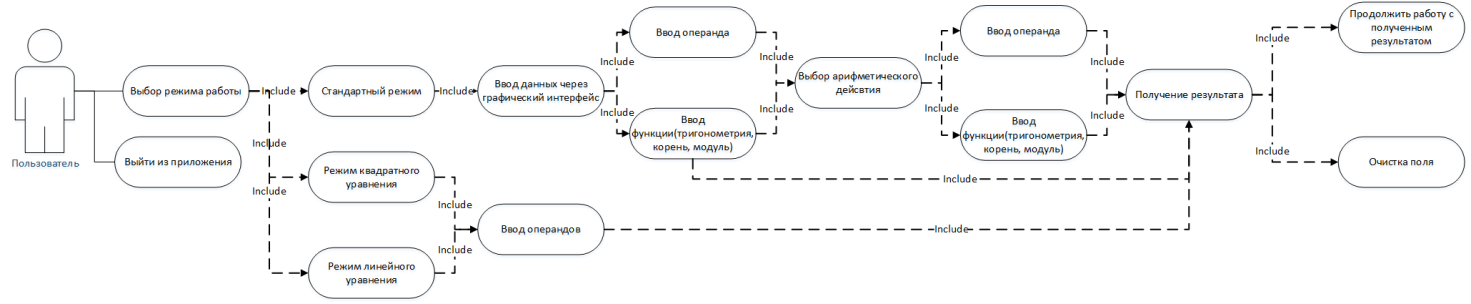
Рисунок 7 – Структурная схема разрабатываемого приложения «Калькулятор».

Рисунок 8 – Диаграмма вариантов использования программы «Калькулятор»

Ввод данных в приложении осуществляется через графический интерфейс с использованием клавиатуры, либо при помощи компьютерной мыши. При этом операндом является целое, либо дробное число, а также математическая постоянная;

При вычислении математических функций (синус, косинус, тангенс, котангенс, двоичный логарифм, модуль) сначала вводится функция, затем операнд.

Для выполнения простых математических операций (сложение, вычитание, деление, умножение, а также возведение числа в степень) производится нажатие соответствующих кнопок графического интерфейса. Затем производится ввод второго операнда.

После ввода выражения, пользователь может получить результат по нажатию кнопки.

После получения результата расчета пользователь имеет возможность продолжить работу с ним, производя следующие математические действия.

Общее назначение программного средства - выполнение арифметических операций для использования в производственном, учебном процессах и повседневной жизни.

Реализуемая задача состоит в том, чтобы при выборе действия выполнялась определенная операция, и имелась возможность сбросить полученный результат или же продолжить расчеты с данным результатом.

**3.2 Используемые функции Win32API**

1)LoadCursor(Instance:NULL,CursorName:IDC\_ARROW) -Загpужает поименованный pесуpс куpсоpа.

* Библиотека:User32.lib
* Параметры:
* Instance**:** Экземпляp модуля, исполнимый файл котоpого содеpжит куpсоp или 0 для пpедопpеделенного куpсоpа.
* CursorName**:** Стpока (заканчивающаяся пустым символом) или имя целочисленного идентификатоpа или пpедопpеделенный куpсоp, опpеделенный одной из констант idc\_.
* Возвращаемое значение:
* В случае успешного завеpшения - идентификатоp куpсоpа; 0 - если куpсоp не найден; не опpеделено, если pесуpс не является pесуpсом куpсоpа.

2)LoadIcon(Instance: NULL,IconName: IDI\_WINLOGO) - Загpужает поименованный pесуpс пиктогpаммы.

* Библиотека: User32.lib
* Параметры:
* Instance**:** Экземпляp модуля, исполнимый файл котоpого содеpжит пиктогpамму или 0 для пpедопpеделенной пиктогpаммы.
* IconName**:** Стpока или имя целочисленного идентификатоpа или пpедопpеделенная пиктогpамма, опpеделенная одной из констант idi\_.
* Возвращаемое значение:
* В случае успешного завеpшения - идентификатоp пиктогpаммы; 0 - в пpотивном случае.

3)RegisterClass(var WndClass: &wc) - Регистpиpует класс окна, атpибуты котоpого опpеделены паpаметpом WndClass, для последующего использования. Класс окна может pегистpиpоваться только один pаз.

* Библиотека:User32.lib
* Параметры:
* WndClass**:** Стpуктуpа **[TWndClass](http://platonov-andrei.narod.ru/Delphi/struct/TWndClass.htm)**.
* Возвращаемое значение:
* Не нуль, если класс заpегистpиpован; нуль - в пpотивном случае.

4)CreateWindow(ClassName: WMAIN\_CLASSNAME, WindowName: WMAIN\_CAPTION,Style: WS\_OVERLAPPED | WS\_CAPTION | WS\_SYSMENU | WS\_MINIMIZEBOX | WS\_MAXIMIZEBOX | WS\_VISIBLE, X: CW\_USEDEFAULT, Y: CW\_USEDEFAULT,Height: 450,Width: 419,WndParent: HWND\_DESKTOP, Menu: NULL,Instance: hInstance,Param: 0) - Создает пеpекpытое, всплывающее или дочеpнее окно.

* Библиотека:User32.lib
* Параметры:
* ClassName**:** Имя класса окна (заканчивающееся пустым символом) или пpедопpеделенное имя класса оpгана упpавления.
* WindowName**:** Заголовок или имя окна (заканчивающееся пустым символом).
* Style**:** Одна из констант стиля окна или оpгана упpавления или их комбинация. К этим константам относятся константы ds\_, ws\_, bs\_, cbs\_, es\_, lbs\_, sbs\_, ss\_.
* X, Y**:** Начальное положение окна или cw\_UseDefault.
* Height**:** Начальная высота окна (в единицах устpойства).
* Width**:** Начальная шиpина окна (в единицах устpойства).
* WndParent**:** Окно владельца.
* Menu**:** Идентификатоp меню или дочеpнего окна.
* Instance**:** Экземпляp соответствующего модуля.
* Param**:** Значение, пеpеданное в [**TCreateStruct**](http://platonov-andrei.narod.ru/Delphi/struct/TCreateStruct.htm) в паpаметpе lParam сообщения [**wm\_Create**](http://platonov-andrei.narod.ru/Delphi/WM_HELP/wm_Create.htm), для создания дочеpнего окна MDI должно быть указателем на стpуктуpу **[TClientCreateStruct](http://platonov-andrei.narod.ru/Delphi/struct/TClientCreateStruct.htm)**.
* Возвращаемое значение:
* В случае успешного завершения - идентификатор окна; 0 - в пpотивном случае.

5)GetMessage(var Msg: &msg,Wnd: NULL,MsgFilterMin: 0,MsgFilterMax: 0) - Считывает сообщение, в pамках диапазона фильтpации, из очеpеди сообщений пpикладной задачи.

* Библиотека:User32.lib
* Параметры:
* Msg**:** Пpинимающая стpуктуpа **[TMsg](http://platonov-andrei.narod.ru/Delphi/struct/TMsg.htm)**.
* Wnd**:** Окно назначения сообщений или 0 для всех окон в пpикладной задаче.
* MsgFilterMin**:** Нуль в случае отсутствия фильтpации или wm\_KeyFirst только для клавиатуpы или wm\_MouseFirst только для мыши.
* MsgFilterMax: Нуль в случае отсутствия фильтpации или wm\_KeyLast только для клавиатуpы или wm\_MouseLast только для мыши.
* Возвращаемое значение:
* Не нуль, если сообщение не **[wm\_Quit](http://platonov-andrei.narod.ru/Delphi/WM_HELP/wm_Quit.htm)**; 0 - в пpотивном случае.

6)DispatchMessage(var Msg: &msg) - Пеpедает в Msg сообщение оконной функции окна.

* Библиотека:User32.lib
* Параметры:
* Msg**:** Стpуктуpа **[TMsg](http://platonov-andrei.narod.ru/Delphi/struct/TMsg.htm)**.
* Возвращаемое значение:
* Значение, возвpащаемое оконной функцией, обычно игноpиpуется.

7)CALLBACK WndProc(Wnd: HWND hWnd,Msg: UINT message, wParam: WPARAM wParam,lParam: LPARAM lParam) - Вызывает и пеpедает инфоpмацию сообщения в адpес пpоцедуpы пpедыдущей функции окна.

* Библиотека:User32.lib
* Параметры:
* Wnd**:** Окно, пpинимающее сообщение.
* Msg**:** Идентификатоp сообщения.
* wParam**:** Дополнительная инфоpмация, зависящая от сообщения.
* lParam**:** Дополнительная инфоpмация, зависящая от сообщения.

8)SendMessage(Wnd: button[26], Msg: BM\_SETCHECK, wParam: BST\_CHECKED,lParam: 0) - Посылает сообщение оконной функции указанного окна. Возвpат из функции осуществляется только после обpаботки сообщения.

* Библиотека:User32.lib
* Параметры:
* Wnd**:** Окно, пpинимающее сообщение или $FFFF для посылки всем всплывающим окнам в системе.
* Msg**:** Тип сообщения.
* wParam**:** Дополнительная инфоpмация о сообщении.
* lParam**:** Дополнительная инфоpмация о сообщении.
* Возвращаемое значение:
* Значение, возвpащенное пpинимающей оконной функцией.

9)GetWindowText(Wnd: hEdit,Str: buf, MaxCount: sizeof(buf)) - Копиpует в Str заголовок окна или текст оpгана упpавления.

* Библиотека:User32.lib
* Параметры:
* Wnd**:** Идентификатоp окна или оpгана упpавления.
* Str**:** Буфеp, пpинимающий стpоку.
* MaxCount**:** Размеp буфеpа Str.
* Возвращаемое значение:
* Фактическое число скопиpованных байт или 0, если текст отсутствует.

10)IsDlgButtonChecked(Dlg: hWnd, IDButton:  ID\_BUTTON\_OP1) - Опpеделяет, отмечен или нет блок упpавления кнопки.

* Библиотека:User32.lib
* Параметры:
* Dlg**:** Идентификатоp блока диалога.
* IDButton**:** Идентификатоp блока упpавления кнопки.
* Возвращаемое значение:
* Не нуль, если отмечен; 0 - если нет. Для кнопок с тpемя состояниями возвpащается 2 - затенен; 1 - отмечен или 0.

11)SetWindowText(Wnd: hEdit,Str: Option1(buf)) - Устанавливает название заголовка для окна или текст оpгана упpавления с помощью стpоки, указанной в Str.

* Библиотека:User32.lib
* Параметры:
* Wnd**:** Идентификатоp окна или оpгана упpавления.
* Str**:** Стpока (заканчивающаяся пустым символом).

12)PostQuitMessage(ExitCode:0) – Напpавляет сообщение [wm\_Quit](http://platonov-andrei.narod.ru/Delphi/WM_HELP/wm_Quit.htm) обычно в ответ на сообщение [wm\_Destroy](http://platonov-andrei.narod.ru/Delphi/WM_HELP/wm_Destroy.htm).

* Библиотека:User32.lib
* Параметры:
* ExitCode**:** Код выхода пpикладной задачи.

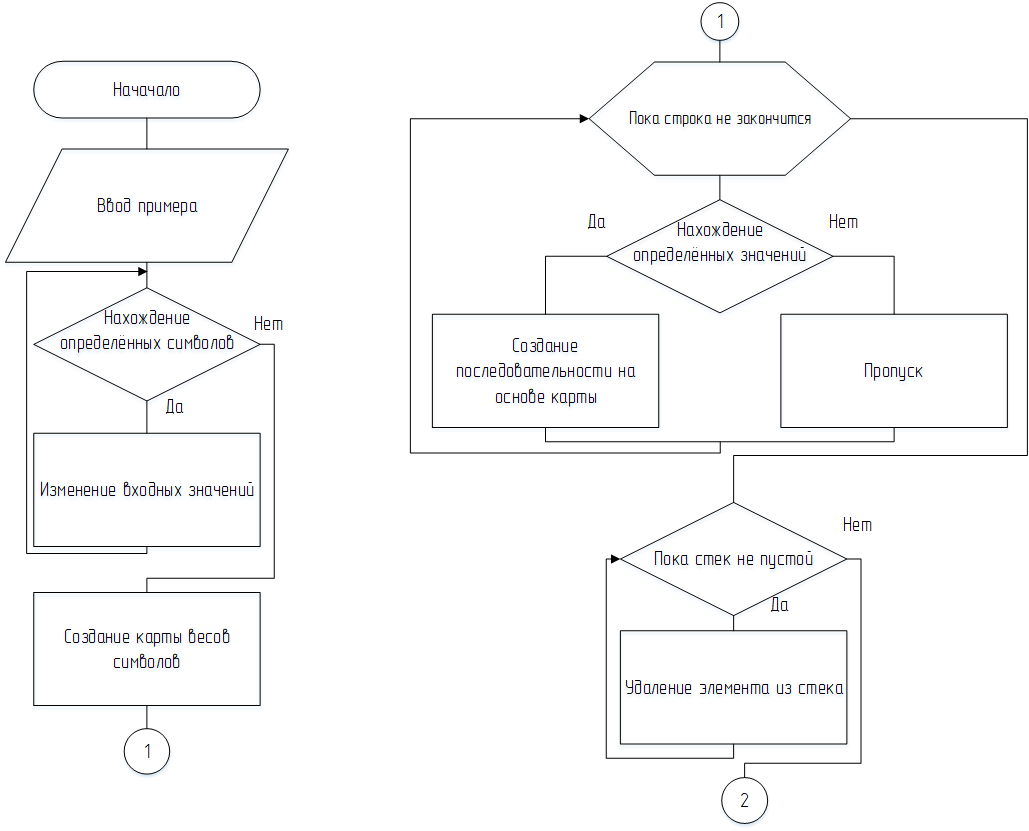
13)DefWindowProc(Wnd: hWnd,Msg: message,wParam: wParam, lParam: lParam) - Обеспечивает стандаpтную обpаботку сообщений для сообщений, котоpые явно не обpабатываются пpикладной задачей.

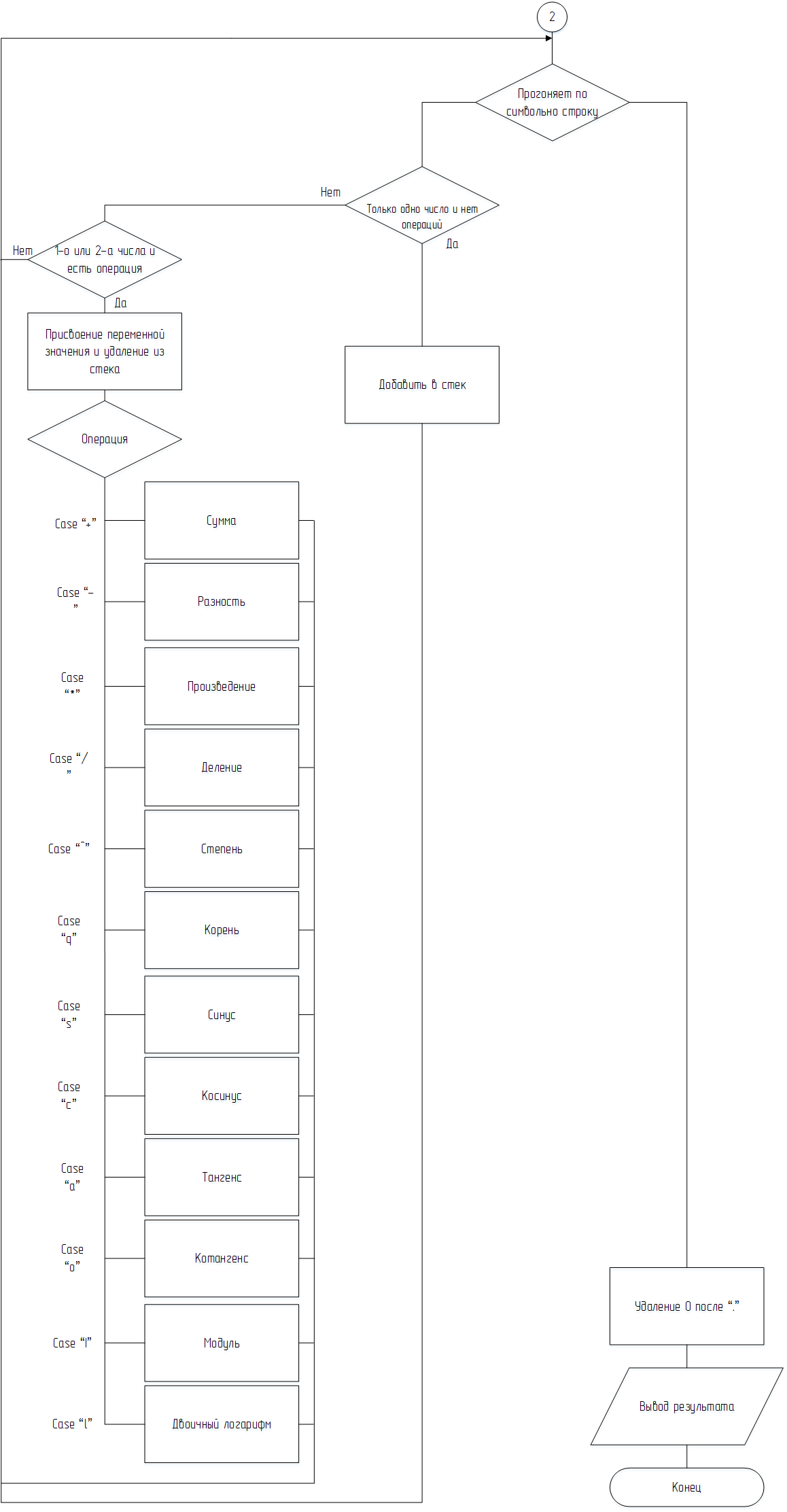
* Библиотека:User32.lib
* Параметры:
* Wnd**:** Идентификатоp окна.
* Msg**:** Номеp сообщения.
* wParam**:** Инфоpмация, зависящая от сообщения.
* lParam**:** Инфоpмация, зависящая от сообщения.
* Возвращаемое значение:
* Результат обpаботки сообщения.

**3.3. Разработка алгоритмов приложения**

Для создания программы калькулятор необходимо реализовать алгоритмы, позволяющие выполнять математические выражения, и получать их результат вычислений, а вычислять линейные и квадратные уравнения. Необходимо организовать возможность сброса полученных результатов.

Блок-схемы алгоритмов представлены на рисунках 9-11.

****

****Рисунок 9 Блок-схема алгоритма обычного режима.(StandardMode)

Принцип работы: Считывает выражение из поля в буфер → редактирует под выполнение карты значений → задаётся карта весов → прогоняет выражение по индексно(в цикле: если находится операционный знак, то задать его последовательность основываясь на карте значений , иначе пропустить) → очищает стек → цикл обработки чисел(если только одно число в поле и нет операций, то добавить его в стек и вывести без изменений, если не одно число или одно , есть операции, удалить из стека и выполнить операцию и записать в стек) , повторять пока не останется одно число в стеке) → проверка , одно число, удаление нулей после точки → вывод результата в поле.

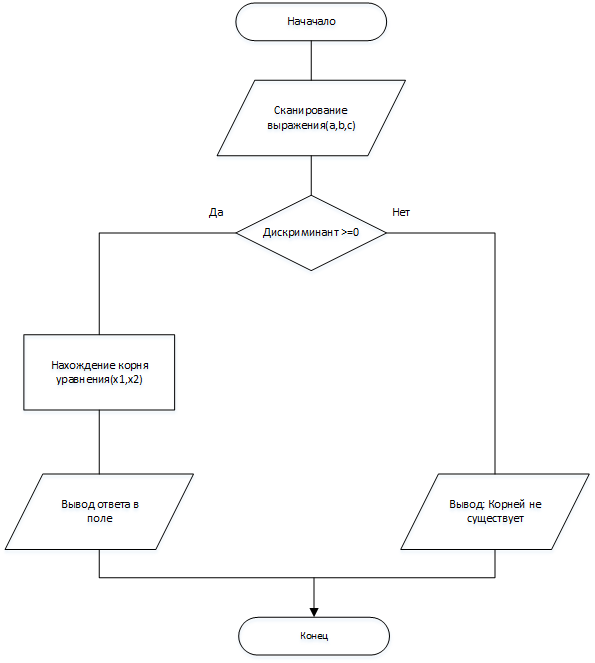


Рисунок 10 Блок-схема алгоритма решения квадратного уравнения.(SquareEquationMode).

Принцип работы: Считывает из поля числа в скобках, проверяется значение дискриминанта. Если дискриминант больше или равен 0, то выполняется расчёт уравнения и вывод его корней в поле. Если дискриминант меньше 0, то выводится «Корней не существует».

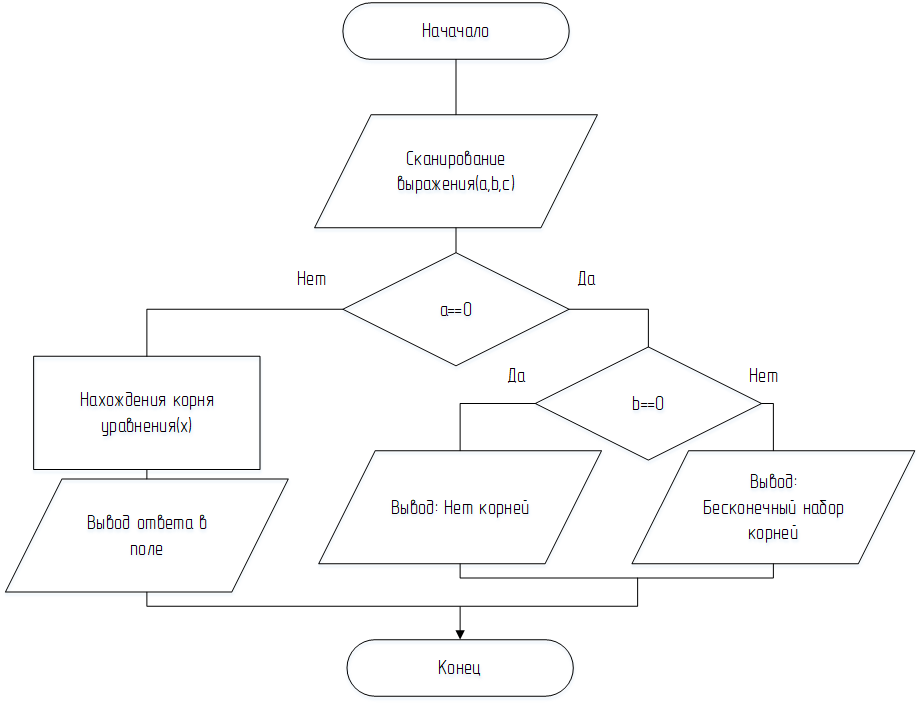


Рисунок 11 Блок-схема алгоритма решения линейного уравнения.(LinearEquationMode)

Принцип работы: Считывает из поля числа в скобках, проверяется условия равенства нулю первого числа. Если первое число не равно 0, то выполняется расчёт уравнения и вывод его корня в поле. Если первое число равно 0, выполняется проверка второго числа на равенство нулю. Если 2-е число равно 0 , то выводится «Нет корней», в противном случае выводится «Бесконечный набор корней»

**3.4. Разработка интерфейса взаимодействия пользователя с системой.**

Приложение разработано для обеспечения работы пользователя, предоставляя возможность быстрого выполнения математических расчетов.

С точки зрения эргономики, в программе создан такой пользовательский интерфейс, который сделает работу комфортной и максимально упрощенной, чем обеспечит удовлетворенность пользователя от работы с программой.

Пользовательский интерфейс (Рисунок 12), состоит из поля(Edit), кнопок взаимодействия с программой(Button) и флагов переключения режимов(Radiobutton).

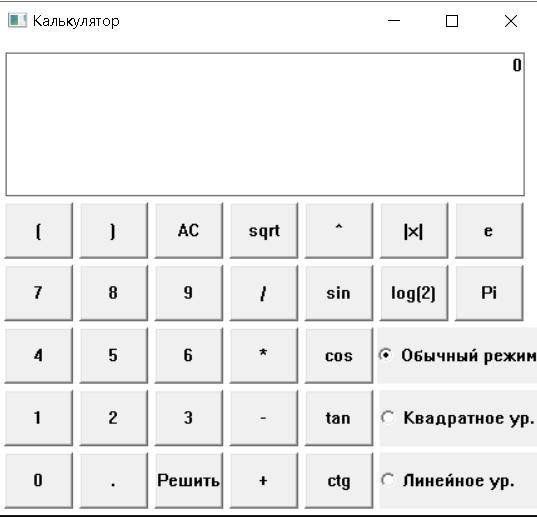


Рисунок 12 – Графический интерфейс основного окна приложения

Для создания окна используется функция:

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam) , где

hWnd – дескриптор окна,

message – сообщение, посылаемое операционной системе,

wParam, lParam – параметры сообщений, для последующего обращения.

В данной функции взаимодействие с окном происходит с помощью множественного условного оператора Switch.

В оператор передается переменная message, в зависимости от значения которой будут выполняться определенные действия.

При message = WM\_CREATE, создается интерфейс взаимодействия пользователя с приложением (окно, кнопки) с помощью встроенной функции CreateWindow.

Message = WM\_COMMAND становится при нажатии пользователем на любую кнопку интерфейса, далее, в зависимости от того, какая именно кнопка была нажата, в программе выполняются определенные действия, например, ввод чисел, знаков, подсчитывание результата.

Если пользователь нажмет на крестик в правом верхнем углу программы, то переменной message присвоится значение WM\_DESTROY и произойдет удаление окна с экрана (программа закроется).

Message = default – это обработчик программы по умолчанию, для того чтобы обеспечить обработку любого сообщения окна, которое приложение не обрабатывает.

**4.Проектно-технологическая часть**

**4.1. Тестирование и отладка макета рабочей программы**

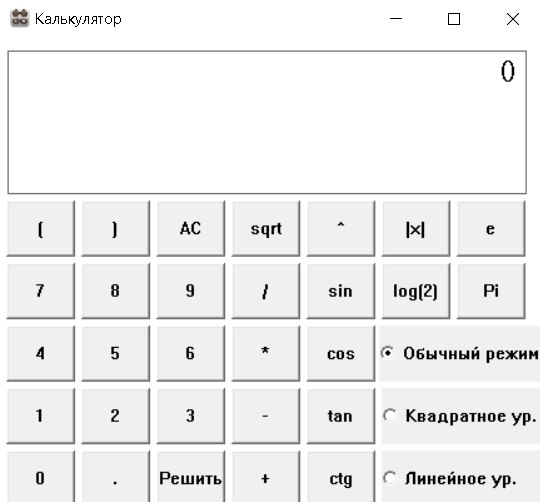


Рисунок 13 – Графический интерфейс основного окна приложения

Экспериментальные данные тестирования процедур и функциональных задач. Материалы тестирования оформляются после описания тестовых примеров в виде таблиц, графиков, снимков экрана с возможным вынесением их в графическую часть курсовой работы. Необходимо также отметить перспективы или факты внедрения разработанного программного приложения.

Для того, чтобы провести тестирование приложения напишем тест-кейс, в который необходимо включить проверку функционала калькулятора, заявленного в техническом задании, а именно: способность графического интерфейса корректно отображать вводимые и получаемые данные, способность вычислять математические выражения, решать линейные и квадратные уравнения. Указанный тест-кейс приведен в таблицах ниже.

Таблица 1. Основная надпись тест-кейса номер 1

|  |  |
| --- | --- |
| **Номер тест-кейса** | 1 |
| **Заголовок** | Осуществить тестирование функционала приложения «калькулятор», заявленного в техническом задании. |
| **Предусловие** | Запустить файл Calculator.exe |

Таблица 2. Тест-кейс номер 1(Тестирование «Обычного режима»)

|  |  |  |
| --- | --- | --- |
| **Номер шага** | **Действие** | **Ожидаемый результат** |
| 1 | Нажать на кнопку «6» графического интерфейса приложения. | На экране интерфейса, в поле, должен появиться символ «6». |
| 2 | Нажать на кнопку «+» графического интерфейса приложения. | На экране интерфейса, в поле, должен появиться новый символ «+» рядом с символ «6». |
| 3 | Нажать на кнопку «9» графического интерфейса приложения. | На экране интерфейса, в поле, должен появиться новый символ «9» рядом с символ «+». |
| 4 | Нажать на кнопку «\*» графического интерфейса приложения. | На экране интерфейса, в поле, должен появиться новый символ «\*» рядом с символ «9». |
| 5 | Нажать на кнопку «8» графического интерфейса приложения. | На экране интерфейса, в поле, должен появиться новый символ «8» рядом с символ «\*». |
| 6 | Нажать на кнопку «Решить» графического интерфейса приложения. | На экране интерфейса, в поле, должен появиться результат вычислений заданного примера, должен быть равен 78. |

Произведем тестирование приложения «калькулятор», по указанному тест-кейсу номер 1.

**Шаг номер 1:**

По нажатию на кнопку «6» главного окна интерфейса программы, в поле, должен появиться символ «6». (Рисунок 14).

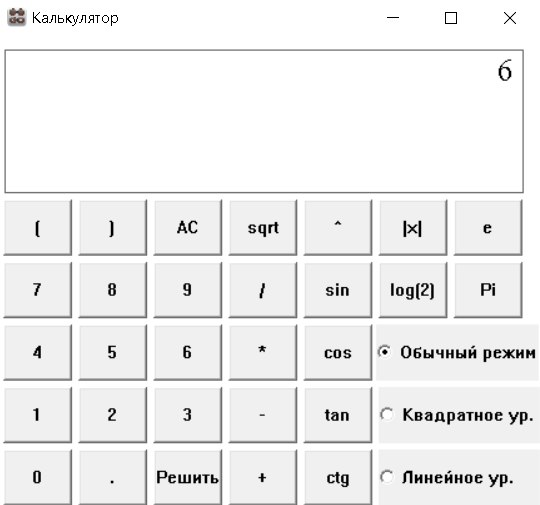


Рисунок 14 – Тестирование приложения, шаг номер 1

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 2:**

По нажатию на кнопку «+» главного окна интерфейса программы, в поле, должен появиться символ «+» рядом с символ «6». (Рисунок 15).

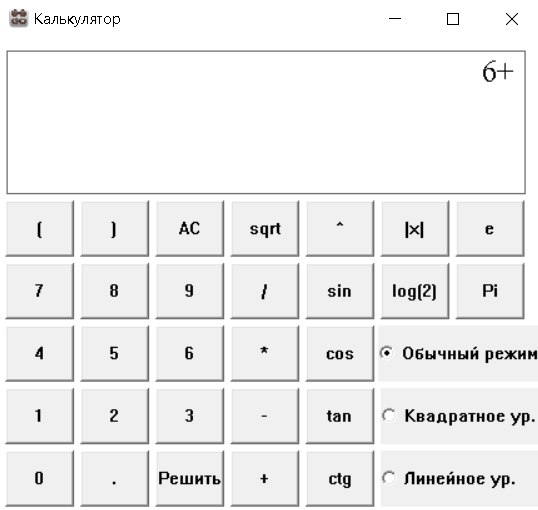


Рисунок 15 – Тестирование приложения, шаг номер 2

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 3:**

По нажатию на кнопку «9» главного окна интерфейса программы, в поле, должен появиться символ «9» рядом с символ «+». (Рисунок 16).

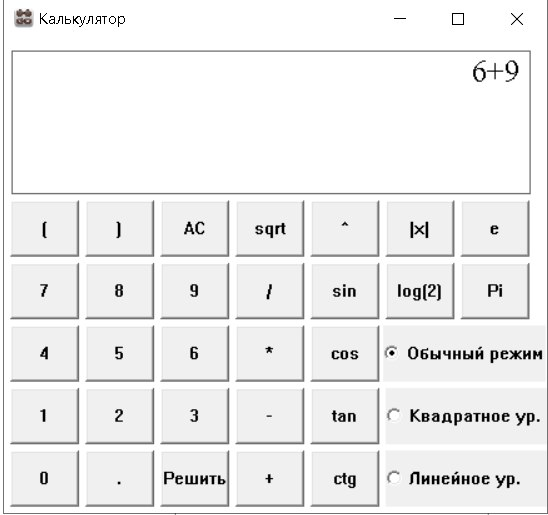


Рисунок 16 – Тестирование приложения, шаг номер 3

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 4:**

По нажатию на кнопку «\*» главного окна интерфейса программы, в поле, должен появиться символ «\*» рядом с символ «9». (Рисунок 17).

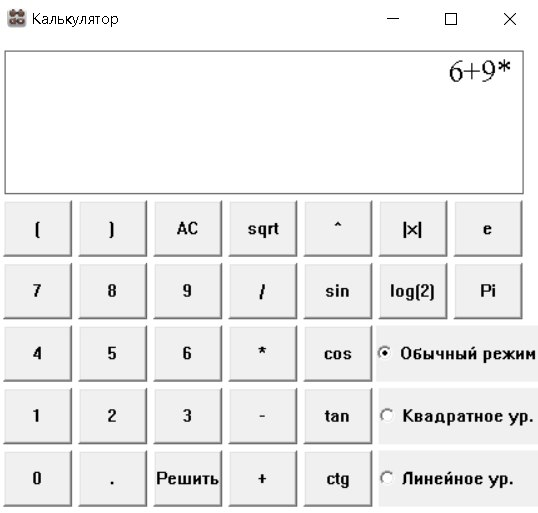


Рисунок 17– Тестирование приложения, шаг номер 4

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 5:**

По нажатию на кнопку «8» главного окна интерфейса программы, в поле, должен появиться символ «8» рядом с символ «\*». (Рисунок 18).

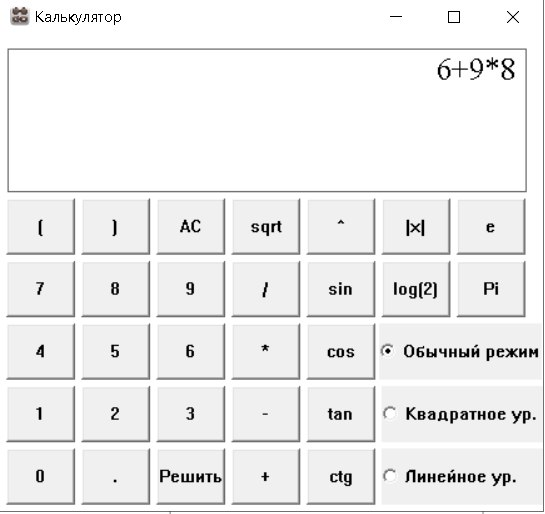


Рисунок 18– Тестирование приложения, шаг номер 5

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 6:**

По нажатию на кнопку «Решить» главного окна интерфейса программы, в поле, результат вычислений заданного примера, должен быть равен 78. (Рисунок 19).

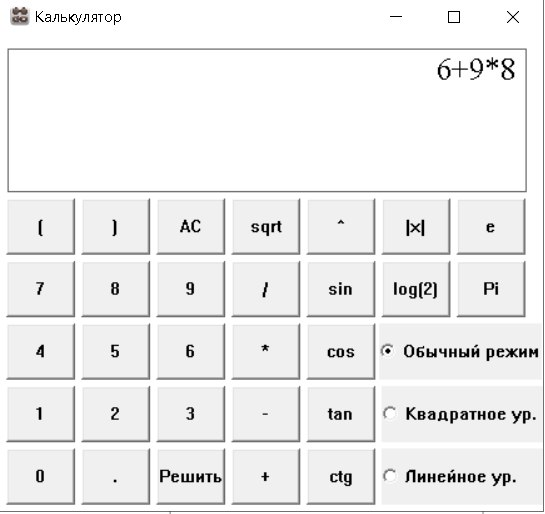


Рисунок 19– Тестирование приложения, шаг номер 6

Вывод: полученный результат совпадает с ожидаемым.

Таблица 3. Тест-кейс номер 1(Тестирование «Квадратное ур.»)

|  |  |  |
| --- | --- | --- |
| **Номер шага** | **Действие** | **Ожидаемый результат** |
| 1 | Нажать на флаг «Квадратное ур.» графического интерфейса приложения. | На экране интерфейса, отобразится переключение, должен появиться флажок около «Квадратное ур.» В поле, должна появится запись «()\*x^2+()\*x+()=0». |
| 2 | В поле графического интерфейса, в первой паре скобок в строке «()\*x^2+()\*x+()=0» ввести «2». | На экране интерфейса, в поле, в первой паре скобок должно появится «2». В поле будет следующее «(2)\*x^2+()\*x+()=0». |
| 3 | В поле графического интерфейса, во второй паре скобок в строке «()\*x^2+()\*x+()=0» ввести «13». | На экране интерфейса, в поле, в третьей паре скобок должно появится «13». В поле будет следующее «(2)\*x^2+(13)\*x+()=0». |
| 4 | В поле графического интерфейса, в третей паре скобок в строке «()\*x^2+()\*x+()=0» ввести «3». | На экране интерфейса, в поле, во второй паре скобок должно появится «13». В поле будет следующее «(2)\*x^2+(13)\*x+(3)=0». |
| 5 | Нажать на кнопку «Решить» графического интерфейса приложения. | На экране интерфейса, в поле, должен появиться результат вычислений заданного примера, который равен равен x1=-6,260399 ; x2=-0,239601. |

Произведем тестирование приложения «калькулятор», по указанному тест-кейсу номер 1.

**Шаг номер 1:**

По нажатию на флаг «Квадратное ур.» главного окна интерфейса программы, в поле, должен появиться строка «()\*x^2+()\*x+()=0». (Рисунок 20).

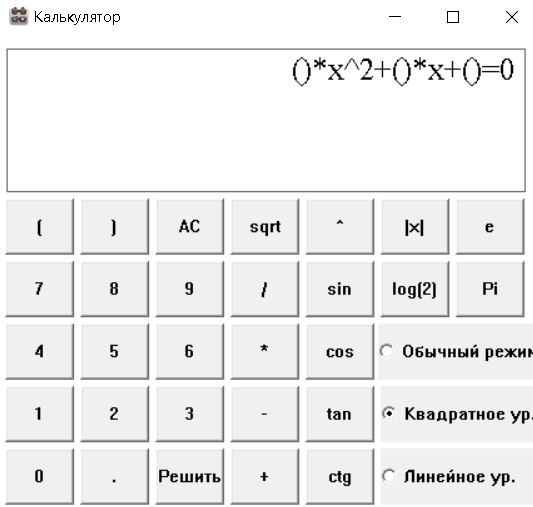


Рисунок 20 – Тестирование приложения, шаг номер 1

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 2:**

В первой паре скобок вводим«2». (Рисунок 21).



Рисунок 21 – Тестирование приложения, шаг номер 2

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 3:**

Во второй паре скобок вводим «13». (Рисунок 22).

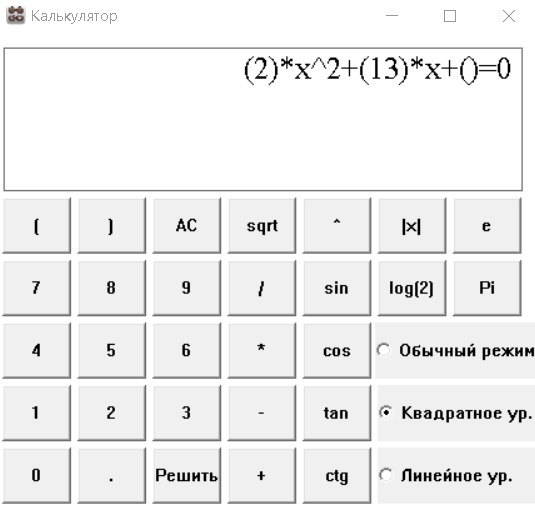


Рисунок 22– Тестирование приложения, шаг номер 3

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 4:**

В третьей паре скобок вводим «3». (Рисунок 23).

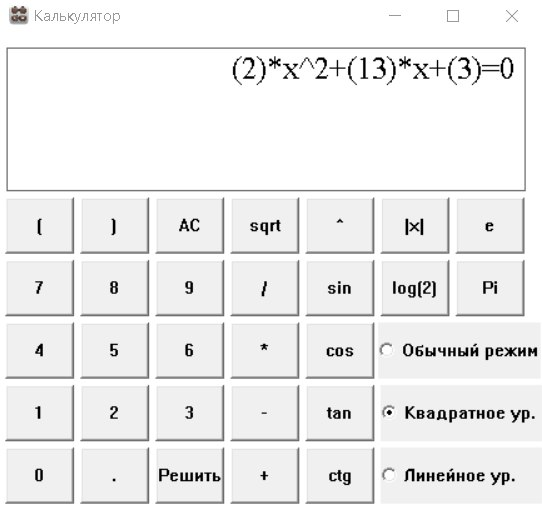


Рисунок 23– Тестирование приложения, шаг номер 4

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 5:**

По нажатию на кнопку «8» главного окна интерфейса программы, в поле, должен появиться символ «8» рядом с символ «\*». (Рисунок 24).

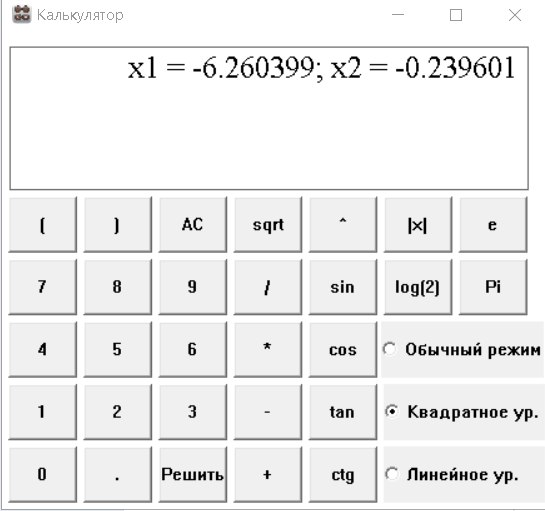


Рисунок 24– Тестирование приложения, шаг номер 5

Вывод: полученный результат совпадает с ожидаемым.

Таблица 4. Тест-кейс номер 1(Тестирование «Квадратное ур.»)

|  |  |  |
| --- | --- | --- |
| **Номер шага** | **Действие** | **Ожидаемый результат** |
| 1 | Нажать на флаг «Линейное ур.» графического интерфейса приложения. | На экране интерфейса, отобразится переключение, должен появиться флажок около «Линейное ур.» В поле, должна появиться запись «()\*x+()=0». |
| 2 | В поле графического интерфейса, в первой паре скобок в строке «()\*x+()=0».ввести «2». | На экране интерфейса, в поле, в первой паре скобок должно появиться «2». В поле будет следующее «(2)\*x+()=0». |
| 3 | В поле графического интерфейса, во второй паре скобок в строке «()\*x+()=0» ввести «13». | На экране интерфейса, в поле, во второй паре скобок должно появиться «13». В поле будет следующее «(2)\*x+(13)=0» |
| 4 | Нажать на кнопку «Решить» графического интерфейса приложения. | На экране интерфейса, в поле, должен появиться результат вычислений заданного примера, который равен x=-6,5. |

Произведем тестирование приложения «калькулятор», по указанному тест-кейсу номер 1.

**Шаг номер 1:**

По нажатию на флаг «Линейное ур.» главного окна интерфейса программы, в поле, должна появиться строка «()\*x+()=0». (Рисунок 25).

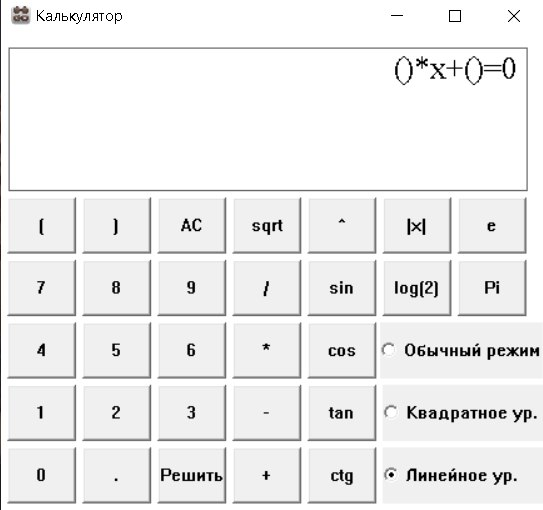


Рисунок 25 – Тестирование приложения, шаг номер 1

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 2:**

В первой паре скобок вводим«2». (Рисунок 26).



Рисунок 26 – Тестирование приложения, шаг номер 2

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 3:**

Во второй паре скобок вводим «13». (Рисунок 27).



Рисунок 27– Тестирование приложения, шаг номер 3

Вывод: полученный результат совпадает с ожидаемым.

**Шаг номер 4:**

В третьей паре скобок вводим «3». (Рисунок 28).

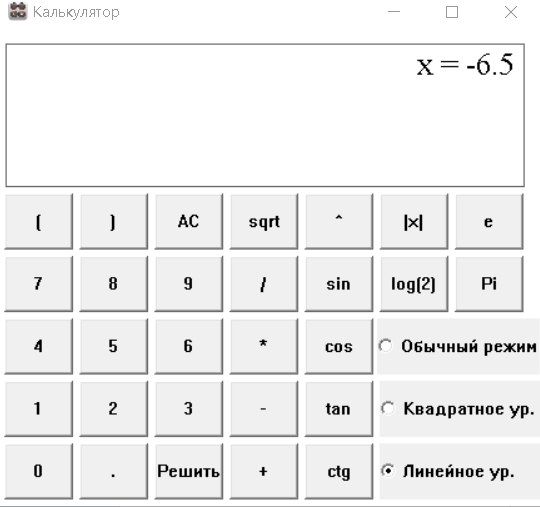


Рисунок 28– Тестирование приложения, шаг номер 4

Вывод: полученный результат совпадает с ожидаемым.

Таким образом, исходя из того, что на каждом шаге полученный результат совпадал с ожидаемым результатом, описанным в тест-кейсе номер 1, можно утверждать, что тестирование функционала приложение прошло успешно.

**4.2. Разработка руководства пользователя и руководства программиста (администратора).**

**Руководства пользователя:**

При запуске приложения калькулятора пользователем должно открываться основное окно графического интерфейса (Рисунок 29), посредством которого пользователь будет взаимодействовать с функционалом разрабатываемого приложения, а именно: расчёт математических операций, решение линейны и квадратных уравнений.

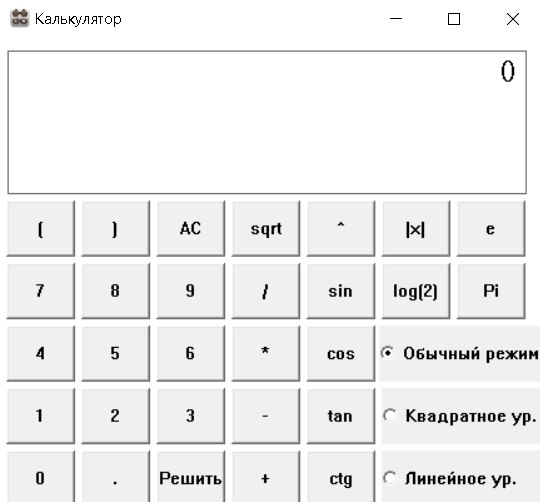


Рисунок 29 – Графический интерфейс основного окна приложения

Входные данные представлены в виде программного кода, который необходимо выполнить при определенных действиях пользователя, а именно:

· нажатие клавиш клавиатуры;

· работа пользователя с кнопками в интерфейсе.

Запуск программы производится двойным щелчком мыши на файле calc.exe или на его ярлыке.

После запуска открывается окно программы. Далее в поле данных мы должны ввести первое число (постоянную величину), затем операцию, которую мы хотим выполнить, а затем второе число. При нажатии кнопки «Решить» получается результат выполнения действия. Его можно сбросить нажатием кнопки «AC», либо продолжить работу с ним.

Перечень элементов интерфейса:

· «AC» – очистка поля;

· «1», «2», «3», «4», «5», «6», «7», «8», «9», «0» – цифры;

· «.” – десятичный разделитель;

· «/» – деление;

· «\*» – умножение;

· «+» – сложение;

· «-« – вычитание;

· «Pi» – постоянная величина (численно равна 3,14);

· «e» – постоянная величина (численно равна 2,71);

· «ctg» – котангенс;

· «sin» – синус;

· «cos» – косинус;

· «tan» – тангенс;

· «log(2)» – двоичный логарифм;

· «sqrt» – квадратный корень;

· «^» – возведение в степень;

· «Решить» – получить результат вычисления.

· «(» и «)» - скобки

· «|x|» - модуль

Перечень режимов:

· «Обычный режим» - стандартный режим работы калькулятора;

· «Квадратное ур.» - режим решения квадратного уравнения;



Рисунок 30 — режим «Квадратное ур.»

В скобках пользователь вводит переменные уравнения. Для перехода в режим «Квадратное ур.», пользователь должен выбрать(нажать) на Radiobutton(флажок) с надписью «Квадратное ур.».

· «Линейное ур.» -режим решения линейного уравнения.

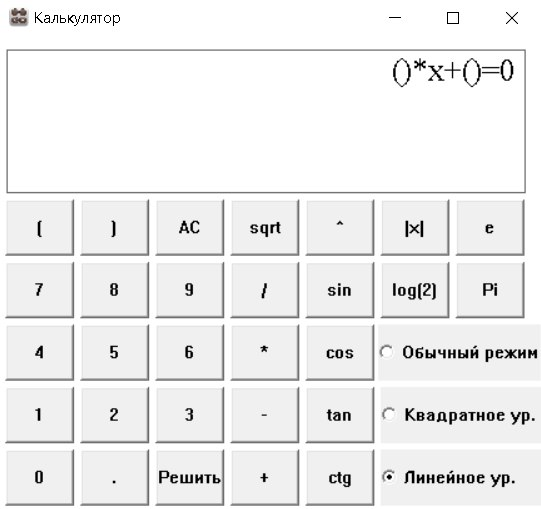


Рисунок 31 — режим «Линейное ур.»

В скобках пользователь вводит переменные уравнения. Для перехода в режим «Линейное ур.», пользователь должен выбрать(нажать) на Radiobutton(флажок) с надписью «Линейное ур.».

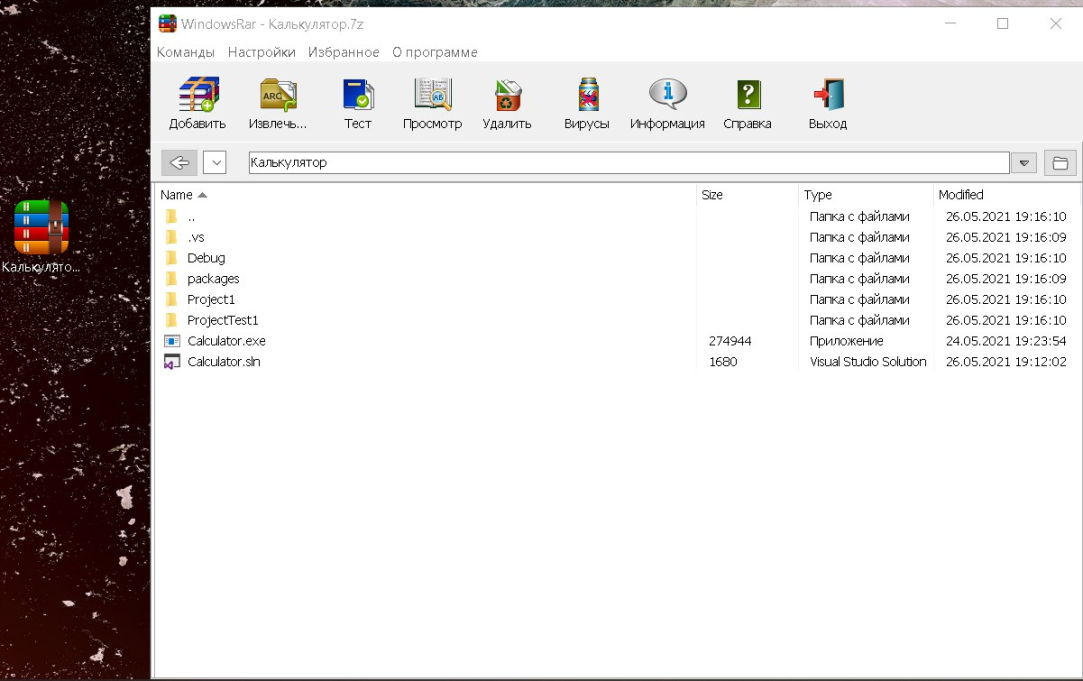
Приложение разработано для обеспечения работы пользователя, предоставляя возможность быстрого выполнения математических расчетов.

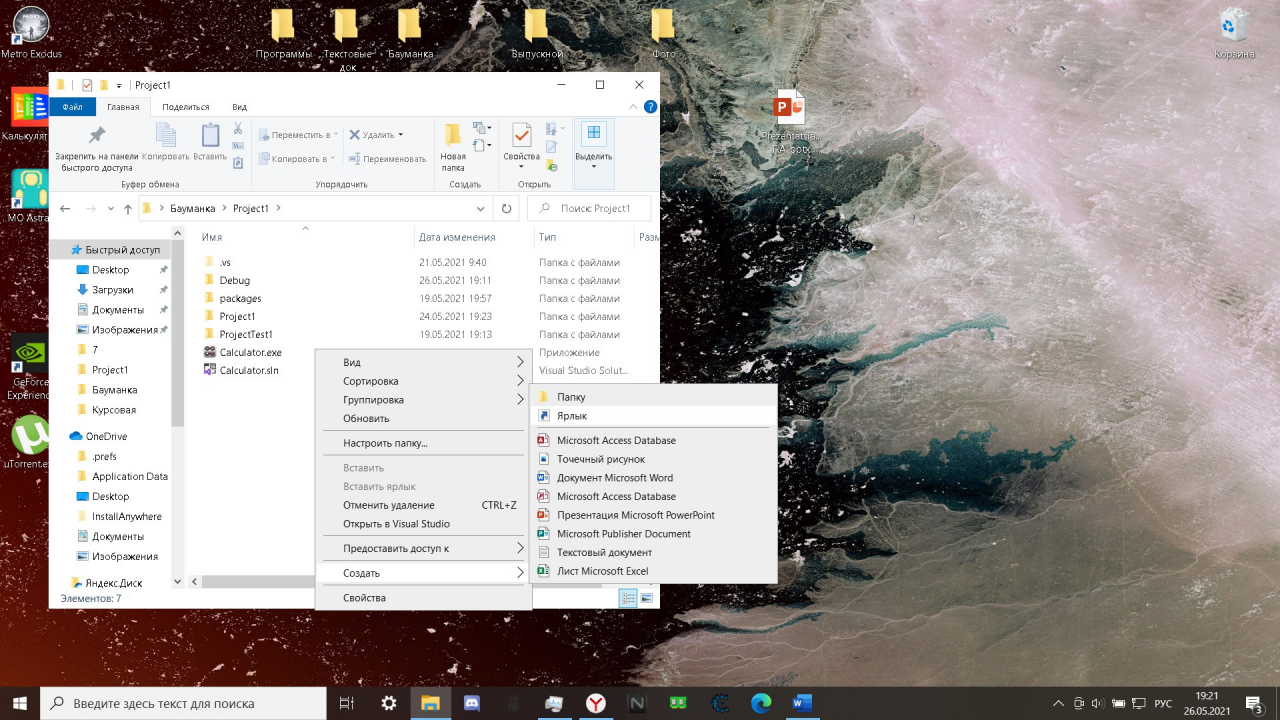
Приложение допускает возможность ввода данных множеством разных способов (клавиатура, мышь).

**Руководства администратора:**

Порядок установки приложения «Калькулятор» состоит из следующих пунктов:

1. cкачать архив с приложением «Калькулятор»;
2. разархивировать архив в удобном месте;

Рисунок 32- разархивация архив

1. создать ярлык приложения на рабочем столе для удобства пользователя.

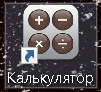
Рисунок 33 - создание ярлыка приложения

Рисунок 34 - ярлык приложения на рабочем столе

После выполнения данных требований установки , пользователь может приступать к работе с приложением

**Заключение**

В ходе выполнения курсовой работы был разработан калькулятор с расширенными возможностями на основе WinAPI.

Работа выполнялась в несколько этапов: была разработана структура системы и реализовано прикладное оконное приложение на основе WinAPI.

Были сформированы навыки по разработке и реализации программного приложения с использованием интерфейса прикладного программирования (АРI) операционных систем.

В данной работе выполнены все поставленные задачи.

В будущем, можно усовершенствовать разработанное приложение путем добавления дополнительного функционала (например, добавление кнопок памяти, записи в/из файла, дополнительные системы счисления).

**ЛИТЕРАТУРА**

1. Павловская Т.А. C/C++ . Процедурное и объектно-ориентированное программирование. Стандарт 3-го поколения. СПб.: Питер, 2015.

2. Платонов В.В. Программно-аппаратные средства защиты информации М.: Академия, 2014.

3. Бречка Д.М. Операционные системы: в 3 ч. Ч. 1. Пакетные файлы и управление компьютером: учебно-методическое пособие. Омск: Омский государственный университет им. Ф.М. Достоевского, 2012 [https://e.lanbook.com/book/75382?category\_pk=1554#book\_name](https://e.lanbook.com/book/75382?category_pk=1554%23book_name)

4. Вирт Н., Гуткнехт Ю. Разработка операционной системы и компилятора. Проект Оберон. М.: ДМК Пресс, 2012 [https://e.lanbook.com/book/39992?category\_pk=1554#book\_name](https://e.lanbook.com/book/39992?category_pk=1554%23book_name)

5. Сидоров В.Н., Сломинская Е.Н., Полникова Т.В., Макарова О.Ю. Оформление графической части выпускной квалификационной работы. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2016.